



## 皮質ネットワークモデル構築のための統合環境開発

著者	我妻 伸彦, 渡辺 哲次, 羽鳥 康裕, 気仙 拓也, 酒井 宏
雑誌名	2007年度CSテクニカルレポート・システム開発型研究プロジェクト特集号
発行年	2007
その他のタイトル	Development of the integrated environment for cortical network model construction
URL	<a href="http://hdl.handle.net/2241/104459">http://hdl.handle.net/2241/104459</a>

## 皮質ネットワークモデル構築のための統合環境開発

我妻 伸彦\* 渡辺 哲次\*\*

羽鳥 康裕\*\* 気仙 拓也\*\*

指導教員 酒井 宏\*\*\*

計算論的モデルの構築とそのシミュレーションは、脳の機能および構造を理解する上で非常に有用である。しかし、現在提案されているニューラルシミュレータは複雑であり、多くの神経科学者にとって使い易いものとはなっていない。そのため、多くの神経科学者が計算論的モデルの有用性を認めているにも関わらず、実際に計算論的モデルを構築しているのはコンピュータサイエンスに精通している一部の研究者だけという現状がある。そこで我々は、ココンピュータサイエンスを専門としていないものでも容易にかつ精緻なモデルを構築でき、さらにコンピュータの機種・OSに依存しないニューラルシミュレータを構築した。これらの要求は、機種に依存せずに使用することができるMATLAB上に関数群を実装することによって実現した。実際に、我々が提案するニューラルシミュレータでシミュレーションを行なう場合には、従来のシミュレータの場合と比較してソースコード長が約90%削減できることを確認した。この結果は、本提案シミュレータは既存のシミュレータが抱えている問題を解決し、計算論的モデルのさらなる普及を促すものであることが期待される。

## Development of the integrated environment for cortical network model construction

Nobuhiko Wagatsuma\*, Satoshi Watanabe\*\*, Yasuhiro Hatori\*\*, Takuya Kisen\*\*, Ko Sakai\*\*\*

Construction of computational models and simulation are helpful for our understanding of brain functions and its organization. However, many neuroscientists tend to avoid constructing the computational model, because of the complexity of current neural simulators (e.g. NEXUS, NEURON). As a result, only some neuroscientists who are familiar with computer science construct the computational model, even if many neuroscientists understand the benefit of the model approach and simulation. Therefore, we develop the novel neural simulator that realizes two demands: (1) easy to construct the precise computational model (2) doesn't depend on machines and OS. To responds those needs, we thought to use the MATLAB that is available on Linux, Windows, Macintosh and implemented set of functions. Actually, we confirmed that when carry out a simulation with our proposal neural simulator, length of source code is reduced by about 90% comparing to the existing neural simulator. This result suggests that proposal neural simulator solves the complexity of existing neural simulator and prompts the dissemination of computational model.

キーワード：ニューラルシミュレータ, Usability, 計算論的モデル

Keywords: Neural simulator, Usability, Computational model

## 1. はじめに

我々の脳は周辺環境の変化を即時に解析し、最適と考えられる解を導いている。例えば、我々は光景の中から物体領

域(図)を瞬時に背景(地)から分離(図/地分離)することができる。しかし、コンピュータに図/地分離を行なわせることは容易ではない。これは図/地分離のアルゴリズムが十分に確立されていないために、コンピュータ上で実現することが難しくなっていると考えられる。図/地分離に限らず、多くの脳機能を工学的に実現するためには、まずヒトの脳を研究し、そのメカニズム・アルゴリズムを明らかにすることが有用な手段と成り得る。

現在、脳の研究においては主に三種類のアプローチが取

\* 筑波大学システム情報工学研究科 CS 専攻博士後期課程

\*\* 筑波大学システム情報工学研究科 CS 専攻博士前期課程

\*\*\* 筑波大学システム情報工学研究科

〒305-8577 茨城県つくば市天王台 1-1-1

Computational Vision Science Labs.,

1-1-1 Ten-nodai, Tsukuba, Ibaraki 305-8577



図 1. 提案システムの処理概要図。ユーザは MATLAB を起動し、シミュレータを立ち上げる。その後、コードファイルをコンパイルすることでモデルのシミュレーションを行う。

られている：生理学的アプローチ、心理学的アプローチ、計算論的アプローチである。その中でも特に計算論的アプローチは脳機能の解明またはその工学的応用のための手段として最も有効である(1)(2)(3)。それにも関わらず、以下に挙げる 2 つの問題がその普及を妨げていると考えられている。

(a) 既存のシミュレータの機種依存性

(b) 計算論的モデルの実装が困難

(a) に関しては、既存のニューラルシミュレータは Linux 環境で開発されているものが多い。そのために計算機の機種依存性も強く、他の OS では起動しないなどの問題が生じる。また、計算論モデルの構築に独自のシステムを導入している研究機関も数多く存在する。これらにより、作成した計算モデルのソースコードやシミュレーションデータが研究機関間で共有がされない等の状況を引き起こす一因となっている。

(b) はコンピュータサイエンス以外を専門とする研究者の参入を阻む主要因になると考えられる。既存のニューラルシミュレータとしては、NEXUS(4)や NEURON(5)が挙げられる。NEXUS は独自の言語体系を、NEURON は C 言語の記述体系を踏襲しており、プログラミング言語に精通したものでないとこれらのシミュレータを扱うことができないという現状を生み出している。そのため、情報工学系を基礎としない生理学、心理学等の研究者が自分の研究成果を基に計算モデルを構築できない、または計算モデルに関心があってもこの研究手法に手を出せないなどが実際に起こっている。

これらの問題を解決した新たなニューラルシミュレータによって、様々な恩恵を得ることが予測できる。例えば、コーディングの時間を短縮できる、モデルのデータベースを作ることによりコードや様々なシミュレーションデータ資源の共有化、また、様々な分野の研究者がモデル研究に参加することでより精緻でわれわれ生体の活動を説明、再現できるモデルを構築することが可能となると考えられる。これらは脳機能に代表される生体のメカニズムの解明だけでなく、そのメカニズムを工学的に応用すると言う意味でも非常に大きな役割を果たす(3)。

## 2. システムの概要

### <2.1> 既存のシステム

神経科学系のシミュレーションを行うためのソフトウェアとして NEXUS(4)、NEURON(5)、そして GENESIS(6)などが挙げられる。これらのニューラルシミュレータは独自のプログラミング言語と GUI による操作を用いることで、ユーザは自分で計算モデルを構築することが出来る。しかし、NEXUS と GENESIS は、Unix 環境下でしか動作しない。その導入やプログラミングのためには UNIX 環境への理解が必要となり、Windows や Macintosh 等の様々なプラットフォームでも動作するシミュレータが望まれている。また、NEURON は Linux、Windows、Macintosh 環境下で動作するシミュレータであるが、NEXUS のように大規模な神経回路のモデル化を行なうことが難しいため、用途が制限されてしまうという問題がある。

### <2.2> 提案システムの利点

前述した既存のシステムが持つ問題点を解決するために、我々は MATLAB(7)を用いて、本システムを構築した(図 1)。MATLAB は Windows、Linux、Macintosh、Sun OS 環境下で動作するプログラミングツールである。そのような MATLAB のライブラリとして、本システムを作成することによって、機種依存性の問題を解決することができる。さらに、MATLAB のライブラリとすることによって、用意された関数だけでなく、ユーザが任意の関数等を作成し、必要に応じて自由にシミュレータを拡張することも可能となる。これにより、1 つの神経細胞の反応を再現する小規模なレベルのモデルから、大脳視覚皮質間のネットワークを研究するような大規模な神経回路のモデル構築にも柔軟に対応することができる。また、様々な規模の計算モデルに適応するために、そのシミュレーション結果の呈示も選択可能とする予定である。

また、プログラミングに精通していないユーザのために、最終的に GUI を用意することを考えている。GUI によって、ユーザはクリックすることでシミュレーション環境を作れるようになるため、プログラミングに関する特別な知識を必要とせずに計算論的モデルの実装を行なうことができるようになる。この機能を実装することで、生理学や心理学のようなプログラミング技術を本格的に持たない研究者も計算モデルを構築することが可能となる。

### 3. システムの実装

ここでは、皮質ネットワークモデルの構築手順について述べる。その後、皮質ネットワーク構築に必要なライブラリを述べる。

#### <3.1>皮質ネットワークの構築手順

皮質ネットワークを構築するためのおおまかな手順は以下の3ステップとなる（図2参照）。

（i）原像の定義

（ii）射影の定義

（iii）写像の定義

まず、（i）では、処理を行うべき対象（原像： $A$ ）の定義を行う。（ii）では、（i）で定義された原像の投影方法（射影： $f$ ）を定義する。（iii）では、（ii）で定義された射影によって得られた結果の投影先（写像： $B$ ）を定義する。以上のような処理を繰り返すことによって、皮質ネットワークを構

（i）原像の定義



（ii）射影の定義

（iii）写像の定義

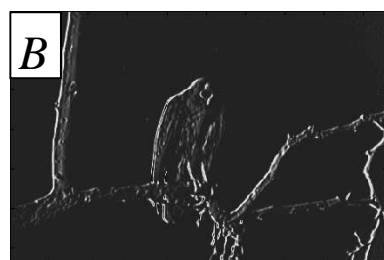


図2 皮質ネットワークの構成例。（i）原像（ $A$ ）を定義する。ここでは、入力刺激としてグレイスケールイメージを定義している。（ii）射影（ $f$ ）を定義する。ここでは、初期視覚野の受容野である Gabor フィルタを射影としている。（iii）写像（ $B$ ）を定義して、最小の皮質ネットワークモデルが構築される。白い部分は細胞が反応している場所を示している。

築していく。つまり、 $f:A \rightarrow B$  を3つのステップによっ

て、実現し、それを繰り返すことによって、皮質ネットワークを構築することができる。

以下に、上述の3ステップを実現するために必要となるシステムの機能や処理について説明する。特に皮質ネットワークモデルを構築するにあたって重要となる事柄について述べる。

#### （i）原像の定義

皮質ネットワークモデルにおける原像は、主に  $xy$  軸で構成される2次元マップである。また、より実環境に近い事柄、現象を扱う場合、2次元マップに時間軸を加えた動画が対象となる。時間軸が加わった場合においても、2次元マップのサイズは変わらないため、ここでは、原像のマップサイズを定義できれば充分である。また、われわれの脳皮質のようにネットワーク間の結合が存在する場合、複数のネットワークが存在する。そのため、ユーザやシステムが各ネットワークを区別・管理するための、名前やネットワークハンドルの定義が必要となる。ただし、ネットワークハンドルは、自動的に割り振られるべきである。

#### （ii）射影の定義

皮質ネットワークモデルにおいて、射影とは伝達関数や処理関数のことを指す。伝達関数として、MATLAB の組み込み関数はもちろん、独自に作成された関数も使用可能にする必要がある。そのためには、原像名と、関数名を指定し、変数を渡す形式が最もシンプルである。また、存在しない原像や関数を参照する等の、ヒューマンエラーを避けるためのエラー表示が必要となる。変数の定義において、射影の定義の際に値を渡す場合と、ファイルを参照する場合の両方に対応する必要がある。このような機能はシミュレーションの高速化や以前のデータを引き継いでシミュレーションを行うために重要な役割を果たす。さらに、変数が望まないデータ形式であった場合のエラー表示が要求される。また、皮質ネットワークモデルでは、しばしば原像が複数ある場合が発生するため、そのようなケースへの対応も求められる機能である。

#### （iii）写像の定義

皮質ネットワークモデルにおける写像は、神経回路の投影先を示す。つまり、入力と出力の接続関係を定義することによってネットワークモデルを構築する。そのために、入力元である原像の名前と、出力先である写像の名前の指定が必要となる。ここでも、存在しない原像や写像を参照した場合のエラー表示が必要となる。また、原像や写像が複数ある場合にも対応できる必要がある。実際、脳皮質には複数の皮質領域から入力を受ける皮質領域が存在している。最後に、原像と写像間の処理関係を伝達関数で定義することで皮質ネットワークモデルが完成する。



図 3. 自然画像中から輪郭を抽出するシミュレーション。(左)入力刺激。画像は The Berkeley Segmentation Dataset より引用。(右)本提案システムの出力画像。ヒトの知覚と一致する輪郭の位置で強い応答を示している。

### <3.2>実装したライブラリ

#### (i)原像の定義関係

##### ●newNetWork(NH, 'NAME', 'SIZE\_X', 'SIZE\_Y')

NH : ネットワークハンドル

NAME : ネットワークの名前

SIZE\_X, SIZE\_Y : ネットワークのサイズ

作成するネットワークの名前とそのサイズの定義を行なう。ネットワーク名とネットワークサイズを定義、ネットワークハンドルを更新して返す。

##### ●loadInputFile(NH, 'INPUT', 'DATA')

INPUT : 入力刺激を読み込むネットワーク

DATA : 読み込む入力刺激

入力刺激の読み込みとそのネットワークへの出力を行なう。ネットワーク名を指定し、入力刺激（強度マップ）を表示する。入力刺激は MATLAB でインポート可能な 1~2 次元の実数配列を対象とした。現段階では、カラーイメージ等の 3 次元配列の入力には対応していないため、3 次元配列を入力とした場合は射影の部分でエラーになってしまう。

#### (ii)射影の定義関係

##### ●newFunction(NH, 'OUTPUT', 'TF', 'ARG')

OUTPUT : 出力ネットワークの名前

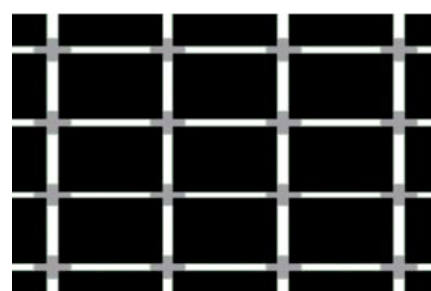
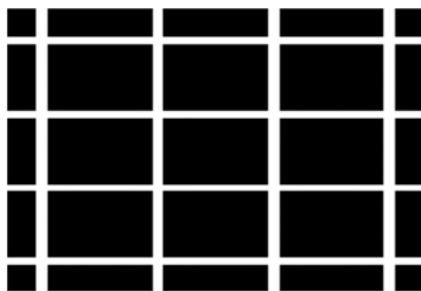


図 4. Herman grid 錯視のシミュレーション。(左)入力刺激。ヒトは入力刺激を見たとき、格子の交差点に存在しない灰色の点を知覚する。(右)本提案システムの出力画像。ヒトの知覚をモデルは再現している。

TF : 伝達関数

ARG : 伝達関数に渡す引数、実数もしくはファイル名で指定することが可能

伝達関数を定義する。ネットワーク名（写像）を指定し、関数名・関数に渡す変数を指定する。

一用意した伝達関数(自由に作成・編集可能、Toolbox を持っているならば使用可能)

▷ linear : 入力をそのまま出力（値域の指定可能である）

▷ sigmoid : シグモイド関数

その他、MATLAB で定義されている関数も使用可能である。

#### (iii)写像の定義関係

##### ●newConnection(NH, 'INPUT', 'OUTPUT')

ネットワークの接続先を定義する。接続元のネットワーク名（原像）と接続先ネットワーク名（写像）を指定することで、システムが皮質間の結合を作成する。

#### (iv)その他

##### ●makeNetworks

本シミュレータシステムの起動、空のネットワークハンドルを返す。

##### ●runNetworks(NH)

シミュレーションの実行、接続関係から実行順序を算出する。算出した順番に従って実行し、各像の結果の表示を行なう。

### <3.3>構築環境

上述の皮質ネットワークモデル構築のために必要な事柄を MATLAB 7.0.0 (R14)を用いて実装した。実装環境は、OS が Microsoft Windows XP Professional Version 2002 Service Pack 2、CPU が Inter® Core™ 2 6700 2.66GHz、2.00 GB RAM である。各種コマンドを作成するに当たって、MATLAB の Toolbox は使用していない。

## 4. システムの評価

構築した本システムを以下の観点から評価した。

(I)既存のシステムを用いる場合とのソースコード長の比較

(II)既存のシステムを用いる場合との処理時間の比較

上記の観点から評価、そして本提案システムが正確に起動するかを検証するために、2種類の計算モデルを作成し、シミュレーションを行なった。

### <4.1>輪郭抽出

初めに、本提案システムが正しく動作することを確認するために、任意の自然画像中から輪郭を抽出するモデルを構築し、シミュレーションを行なった。その結果の一例を図3に示す。図3左は入力刺激として用いた自然画像で、The Berkeley Segmentation Dataset(8)に含まれている画像である。図3右は伝達関数として、前章で紹介した”sigmoid”を用いた場合の本提案システムの応答である。図3右の白い部分が強

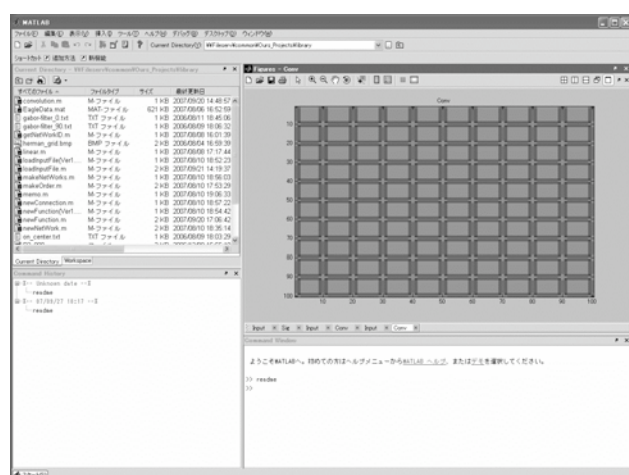


図 5. 本システムを用いてシミュレーションを行い、結果の描画が終わったときに出力されるウィンドウ。この画面が表示されたときに、シミュレーション終了とする。

い反応を示している領域であり、黒い領域は反応が弱い、もしくはないことを示す。われわれヒトの知覚と比較しても、輪郭の位置で、モデルの反応が大きくなっていることがわかる(e.g. 木や鷺の左肩)。この結果はシミュレーションが正しく行なわれていることを示しており、我々が実装したシステムが正しく動作していることを示すものである。

### <4.2>Herman grid 錯視

次に、既存のシステムと本提案システムの性能を比較するために、Herman grid 錯視のシミュレーションを既存のシミュレーションシステムと本提案システムの両方で実装した。このモデルはヒトの初期視覚の処理を再現する計算モデルである。本提案システムにより構築された初期視覚皮質モデルのシミュレーション結果を図4に示す。図4左は入力刺激であり、図4右はこのモデルの応答である。ヒトは格子の交差点上に灰色の点を知覚するが、このモデルはヒトの知覚特性を良く再現している。

(I)ソースコード長の比較

このモデルは以下の7行のコマンドを実行することで作成した。

```
>N=makeNetWorks;  
>N=newNetWork(N,'Input','100','100');  
>N=newNetWork(N,'Conv','100','100');  
>N=loadInputFile(N,'Input','herman.bmp');  
>N=newFunction(N,'Conv','convolution','on_center.tx  
t');  
>N=newConnection(N,'Input','Conv');  
>N=runNetWorks(N);
```

同様のシミュレーションを既存のニューラルシミュレータ NEXUSで行なう場合には52行もソースコードを記述する必要がある。従って、本提案システムと NEXUS を比較した場合、本提案システムは約 90% の効率化に成功したことになる。

(II)処理時間の比較

NEXUS を動かしたマシンのスペックは Redhat Linux 9.0、Intel® Itanium®, 733MHz, 300MB RAM である(提案システムを動かしたマシンのスペックは 3.3 を参照)。Herman grid 錯視のシミュレーションを行い、描画が終わるまで(図5)にかかる時間を比較したところ、NEXUS では約 5 秒かかり、提案システムの場合には 0.078 秒かった。マシンのスペックが異なっているので単純な比較をすることはできないが、シミュレーションに必要な時間が 1/70 程度になっていることから、高速化に成功しているといえるだろう。

以上のことから、ソースコード長・処理時間のどちらにおいても、既存のシミュレータより優れた結果を本提案システムは示した。加えて、たった7行のソースコードでヒトの知覚を再現するモデルを構築することができるシミュレータ



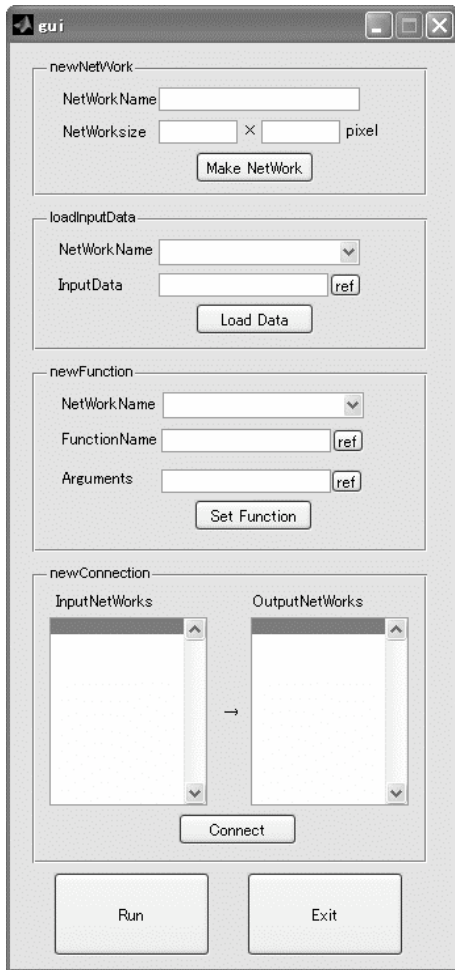


図 6. 一部実装が完了している GUI。ユーザは GUI 上でパラメータの変更や、ネットワークの接続関係の定義を行うことができる。

は他に存在しておらず、本提案シミュレータは神経科学の研究に計り知れない影響を与えることが容易に予想される。

## 5. 終わりに

本プロジェクトでは既存のニューラルシミュレータ・システムが持つ問題点を分析し、既存のものとは全く異なる計算モデルのシミュレーションシステムを構築した。具体的には、機種や OS に依存せず、プログラミングに精通しない研究者であっても容易にモデルの実装を行なうことができるシステムである。このようなシステムを構築するために、多様なプラットフォームで動作する MATLAB 上に、大脳皮質ネットワークモデルを構築するために必要となるライブラリを実装するという方法を用いた。ライブラリが存在することによって、ユーザは特定のコマンドを実行するだけでモデルの実装を行なうことができるので、特にプログラミングに精通していない研究者は複雑なプログラミングの作業から解放されることが期待される。また、プログラム技術を持つ

研究者であっても、本提案システムを用いることにより、複雑なコーディング作業から解放される可能性が示唆された。

実際に、実装したシステムを用いて初期視覚皮質モデルのシミュレーションを行なったところ、わずか 7 行のソースコードでヒトの知覚を再現することができた(既存のシミュレータの場合は 52 行)。実行時間も 0.078 秒であり、即座にシミュレーションの結果を得ることができた(既存のシミュレータの場合は約 5 秒)。この結果は本提案システムの実装効率・計算効率の良さを示すものであり、ユーザの心理的・物理的負担が大いに軽減されることを期待できる。

本提案システムは既存のシミュレータが持つ機種依存性、モデル実装の複雑性という課題を解決し、さらに計算効率も良いという特徴を持っている。誰でも容易に計算論的モデルを構築することができる本提案システムは、神経科学のさらなる進展を促すに留まらず、計算論的アプローチの有用性を再確認するためのツールとして機能するであろう。

本提案システムの今後の拡張としては次のものを考えている。

- ・ GUI の実装
- ・ 一对多の入出力関係への対応
- ・ ネットワーク構造の可視化
- ・ ネットワーク間の対応関係の可視化

以下、それぞれの拡張項目について述べる。

実装予定の GUI を図 6 に示す。各パネルの意味は以下の通りである。

### ●newNetWork：新規ネットワーク（原像）の作成

- ▷NetWorkName...ネットワークの名前の定義
- ▷NetWorkSize...縦、横の順にピクセルサイズで指定
- ▷Make NetWork...上記 2 つを入力したら、作成されたネットワークが表示される

### ●loadInputData：データをロードして表示

- ▷NetWorkName...データをロードするネットワークの指定 (リストボックスにしたのは、ユーザが存在しないネットワークを指定しないため)
- ▷InputData...ロードするデータの指定、カレントディレクトリに MAT ファイルがある場合は名前を指定するだけでよい、他のディレクトリのデータを指定したい場合は ref で参照

▷Load Data...上記 2 つを入力したら、データをロードして、指定したネットワークに表示する。InputData で指定したデータが存在しない場合はエラーを返す。

### ●newFunction：新規伝達関数（射影）の指定

- ▷NetWorkName...伝達関数を設定するネットワークの指定 (リストボックスにしたのは、ユーザが存在しないネットワークを指定しないため)
- ▷FunctionName...設定する伝達関数の名前の指定、カレントディレクトリに m ファイルがある場合、または MATLAB の組関数を使用する場合は名前を指定するだけで

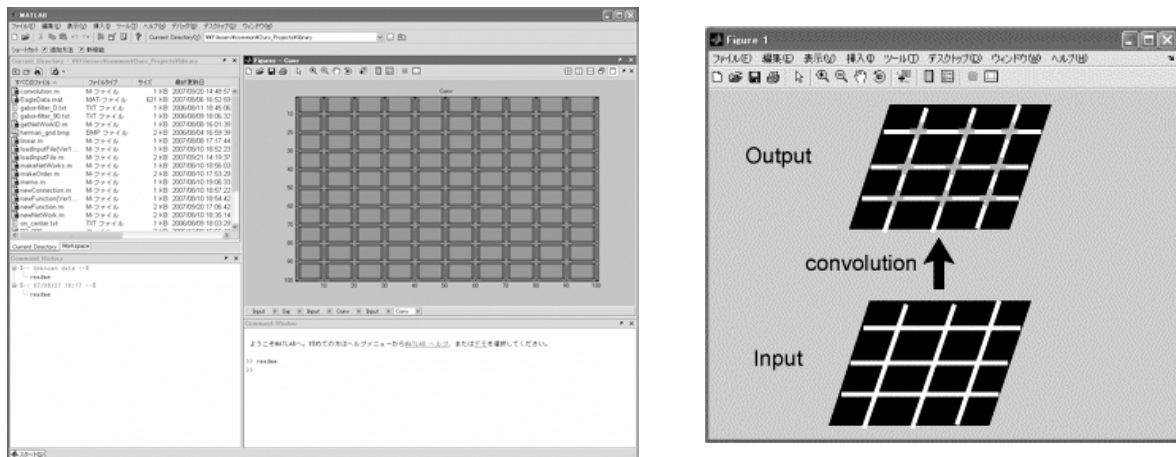


図7. ネットワークの構造を可視化する機能のイメージ図。(左)MATLAB上でシミュレーションを行なった際に表示される画面。(右)左のシミュレーションを行なったときに作成したネットワークを可視化したもの。ネットワークの構造を可視化するにはネットワーク名、伝達関数、接続元と接続先、各種パラメータを見ることができる。

よい、他のディレクトリのデータを指定したい場合は **ref** で参照

▷**Arguments...** 設定する伝達関数に渡したい引数の定義、ファイルを参照したい場合は **ref** で参照

▷**Set Function...** 上記3つを入力したら、指定したネットワークに伝達関数を設定する。存在しない **Function** や、引数の指定違反がある場合はエラーを返す。

#### ●**newConnection** : 接続元と接続先 (写像) の指定

▷**InputNetWorks...** 存在するネットワークから接続元のネットワークを選択する。リストボックスにしたのは複数のネットワークを指定できるように。

▷**OutputNetWorks...** 存在するネットワークから接続先のネットワークを選択する。リストボックスにしたのは複数のネットワークを指定できるように。

▷**Connect...** 指定した接続元と接続先を設定する。

#### ●**Run** : シミュレーションの実行、描画

#### ●**Exit** : シミュレータの終了

ユーザは **GUI** 上でパラメータや伝達関数、ネットワーク間の接続関係を変更することができるようになるため、本提案システムの利便性をさらなる向上が期待できる。加えて、コマンド群を呼び出さずにモデル構築を行なうことも可能となるため、計算論的アプローチへの導入として本提案システムを用いることにより、今まで計算論的アプローチになじみのない、またはこの研究手法に関心を持っている研究者に対して門戸を広げることも可能となる。

現在のシステムでは、ネットワーク間の関係を一对一に制限している。しかし、脳内ではネットワーク間の対応関係

は一对多ということが多くの研究から示されている。複数のネットワーク間の結合を記述することができるようにシステムを拡張することによって、大脳皮質のより精緻なモデル化を行なうことが可能になる。また、大脳皮質はフィードフォワードだけでなくフィードバック回路も重要な役割を果たしていることが知られている。従って、本提案システムにおいてもフィードバックを実現することができるように拡張する必要があると考えている。

モデルが複雑になるほど、ネットワークの数は増え、その対応関係は複雑になる。そうなる、自分の意図した通りにモデルが実装されているのかということやネットワークの構造がどうなっているのかを把握することが難しくなってしまう。ネットワークの構造を可視化する機能によって、ユーザは容易に構築しているネットワークを確認することができる(図7)。ネットワーク構造を可視化することによって、ユーザはネットワーク名、伝達関数、ネットワークの接続関係(接続元と接続先)、ネットワークのサイズなど、ネットワーク構造を確認する際に特に重要となる項目を目で確認することができる。この機能によって、ヒューマンエラーの大幅な減少が期待される。

ネットワーク間の対応関係の可視化というのは、前述のネットワークの構造の可視化よりもミクロな単位での可視化を意味している。例えば、写像の任意の点をクリックすると、現像のどの点から入力を受けているのか、どのような写像が用いられているのかを表示するような機能を想定している。この機能により、ソースコードを辿ることなくモデルを理解することも可能になるため、ソースコードの共有などが促進されることを期待することができる。

以上のような課題を解決することでユーザの利便性を追求していくことを考えている。また、実際に本提案システム



---

のユーザと成り得るであろうと想定される、国内外の研究機関・大学の研究者の意見を取り入れることによって、より汎用的で利用価値のあるシステムを構築していく。

## 文 献

---

- (1) K. Sakai and H. Nishimura, “Surrounding suppression and facilitation in the determination of border ownership”, *Journal of Cognitive neuroscience*, p562-p579, 18, 2006
- (2) N. Wagatsuma, R. Shimizu and K. Sakai, “Effect of spatial attention in early vision for the modulation of the perception of border-ownership”, *ICONIP2007*, 2007
- (3) 我妻伸彦, 辻 義尚, 成島和樹, 渡辺哲次, 酒井 宏, “実画像に対する人間の知覚特性に基づく図方向知覚を再現するシミュレーションシステムの構築”, 2006年度システム開発型研究プロジェクト
- (4) P. Sajda, K. Sakai and L. H. Finkel, “NEXUS: A tool for simulating large-scale hybrid neural networks”, *Proceedings of the Summer Computer Simulation Conference*, p72-76, 1992
- (5) N. T. Carnevale and M. L. Hines, “The NEURON Book”, CAMBRIDGE UNIVERSITY PRESS, 2006
- (6) J. M. Bower and D. Beeman, “The Book of GENESIS”, Springer-Verlag, 1998
- (7) MATLAB: The Math Works
- (8) The Berkeley Segmentation Dataset  
<http://www.eecs.berkeley.edu/Research/Projects/CS/vision/grouping>